**Natural Language Processing HW1 -** *submitted by Iron Code Benders*

[*Github*](https://github.com/adeb567/CSCI-5541-Spring-HW1)

• Whether you replicated the classifier from the original author’s code (Option 1) or fine-tuned it using HuggingFace (Option 2)

Option 2: Fine-tuned "distilbert-base-uncased" using HuggingFace

• Description of the task and models with references to the original papers and model cards/re-

Pository.

Task - Sentiment Classification

Sentiment classification is the automated process of identifying opinions in text and labeling them as positive, negative, or neutral, based on the emotions customers express within them.[[Link1](https://monkeylearn.com/blog/sentiment-classification/#:~:text=Sentiment%20classification%20is%20the%20automated,emotions%20customers%20express%20within%20them.)]

For our task we have used the SST2 dataset.

The Stanford Sentiment Treebank is a corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language. The corpus is based on the dataset introduced by Pang and Lee (2005) and consists of 11,855 single sentences extracted from movie reviews. It was parsed with the Stanford parser and includes a total of 215,154 unique phrases from those parse trees, each annotated by 3 human judges.

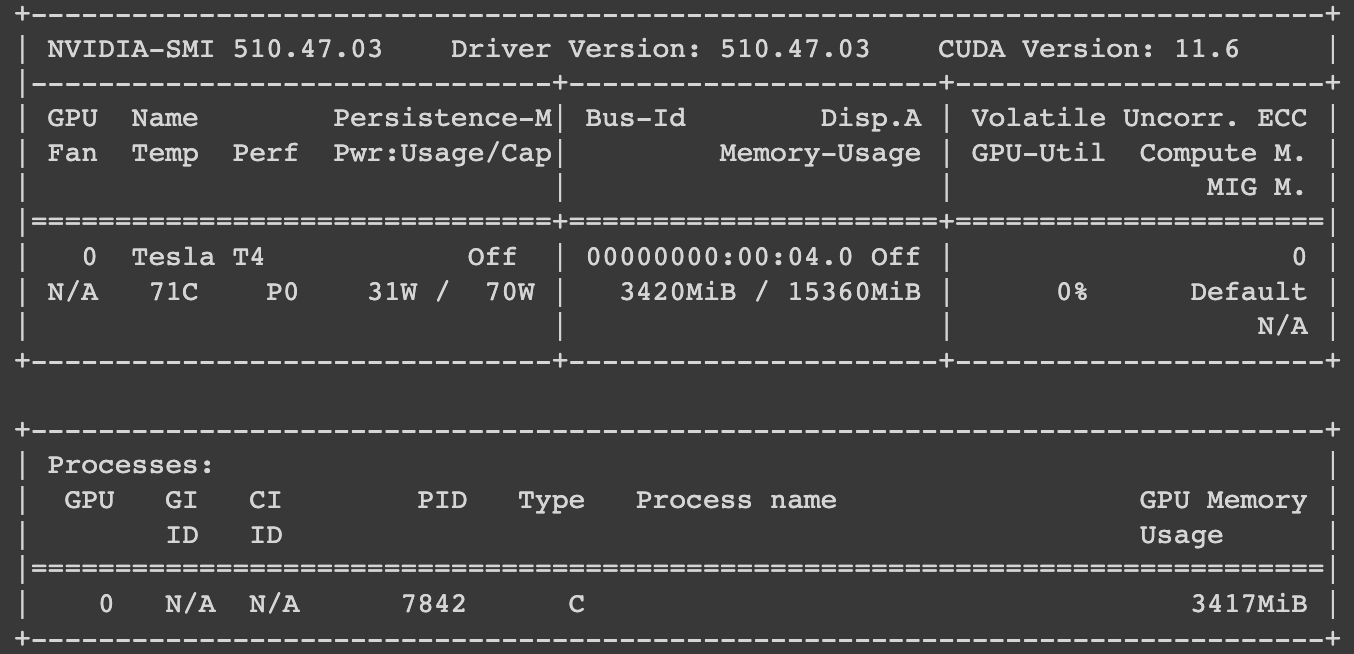
Binary classification experiments on full sentences (negative or somewhat negative vs somewhat positive or positive with neutral sentences discarded) refer to the dataset as SST-2 or SST binary.[[Dataset](https://aclanthology.org/attachments/D13-1170.Attachment.pdf)][[Link2](https://huggingface.co/datasets/sst2)]

BERT is a transformers model pre-trained on a large corpus of English data in a self-supervised fashion. This means it was pretrained on the raw texts only, with no humans labeling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts.[[BERT](https://arxiv.org/abs/1810.04805)]

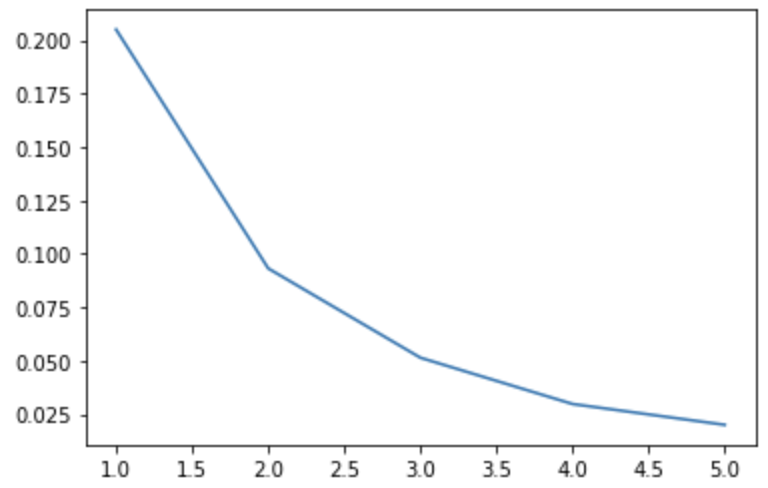


• What kind of hardware you ran your model on

Google Colab GPU



• How do you ensure your model has been trained correctly? Do you have a learning curve graph of your training losses from forward propagation? What does it look like?



We can see that the training loss decreases with the number of epochs. This shows that that the predictions after every epoch are closer to the labels and hence we can conclude that the model is learning.

• Evaluation metrics used in your experiment

The evaluation metrics used in our experiment are -

Training loss, Evaluation loss - We have used the logits loss function for our trainer. The logits loss, also known as cross-entropy loss, is a measure of the difference between the predicted probabilities and the true labels in a classification problem. The goal of training a neural network is to minimize this loss function. The logits loss is defined as:

Loss = -∑(y \* log(p) + (1 - y) \* log(1 - p))

where y is the true label (0 or 1), p is the predicted probability of the positive class, and the summation is over all the classes.[ChatGPT]

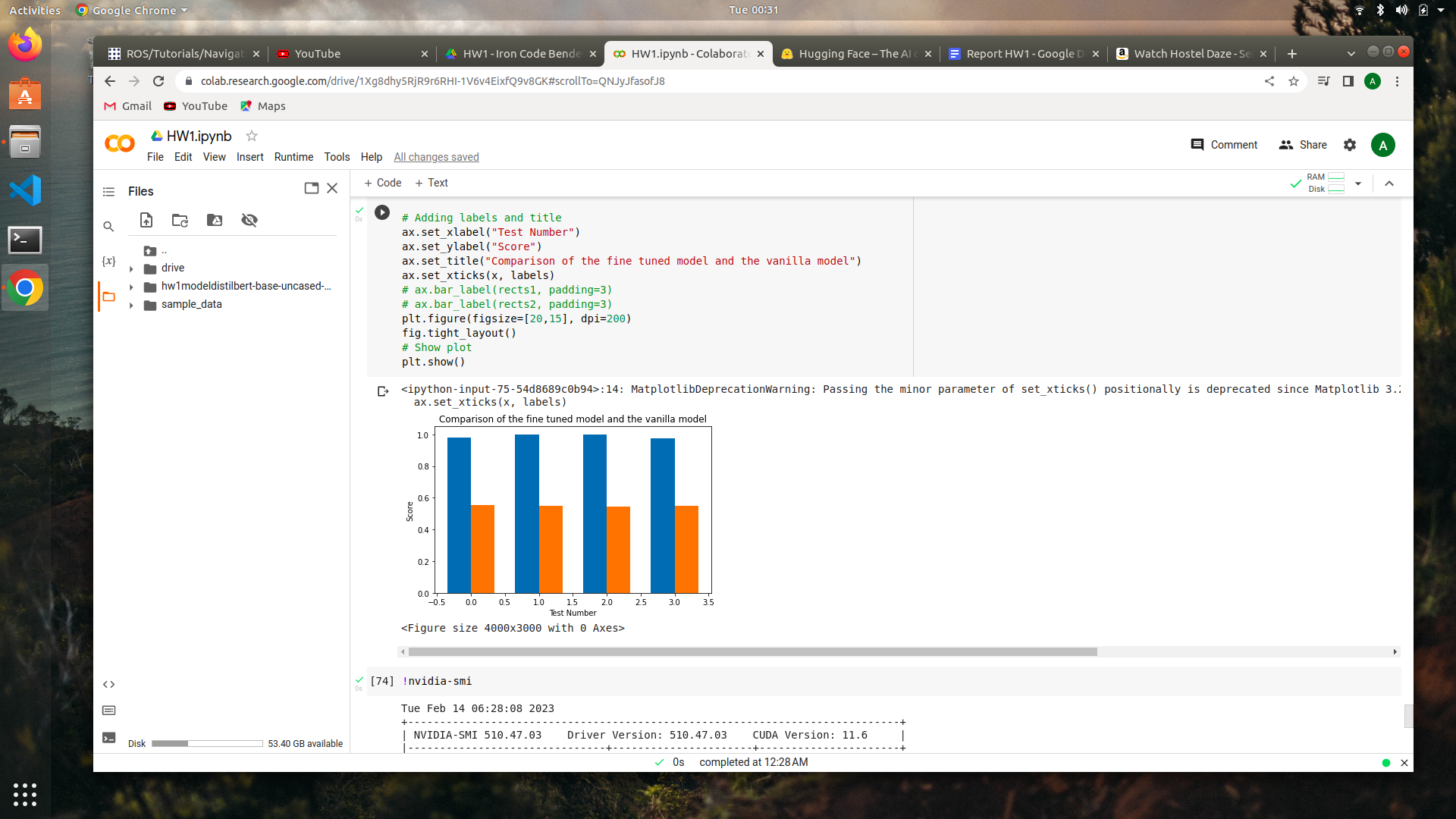
Training Accuracy, Evaluation accuracy - Accuracy is the number of correct predictions out of total predictions in the corresponding data set.

• Test set performance and comparison with score reported in original paper or leaderboard. A

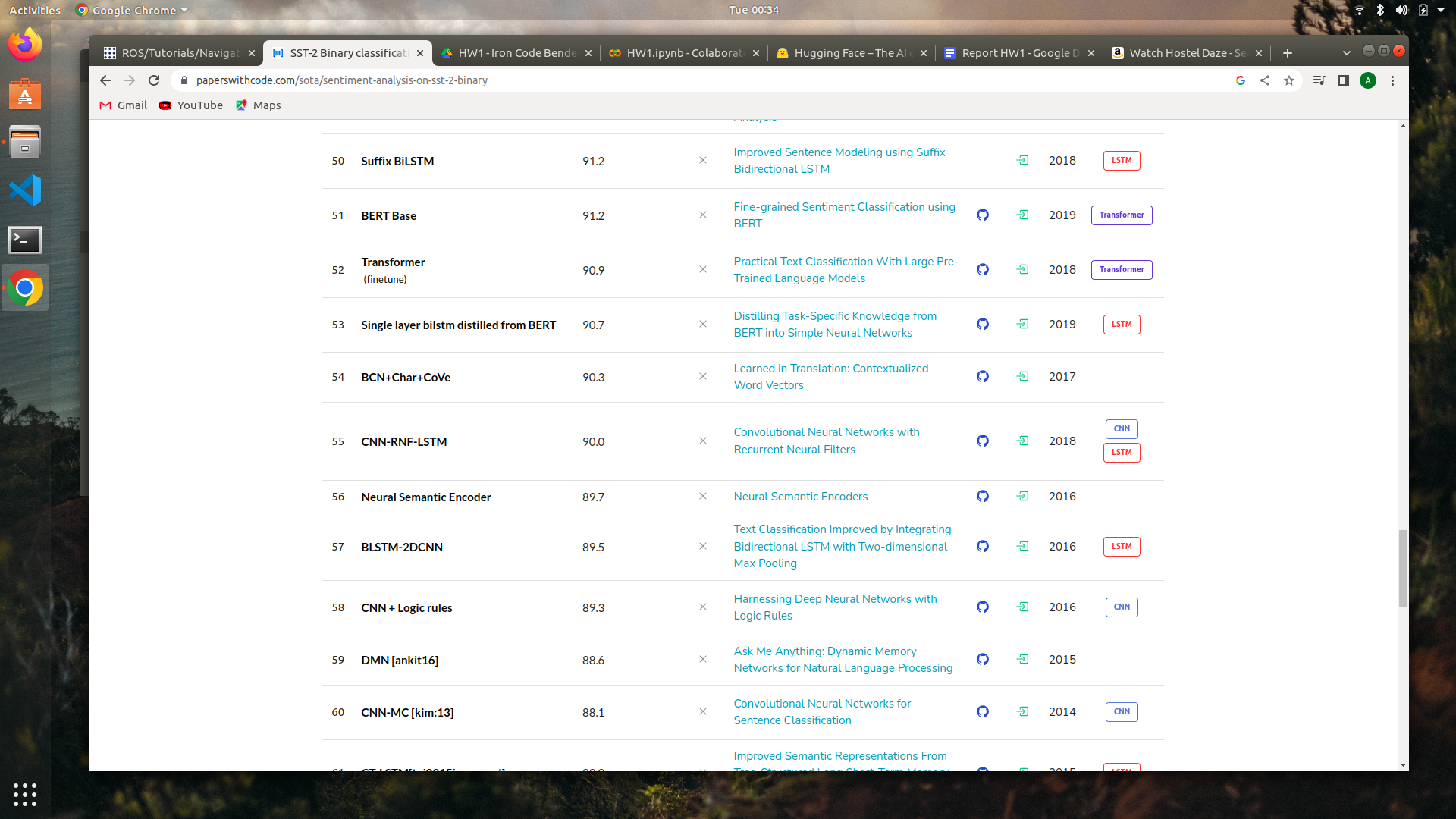
justification is needed if it differs from the reported scores.

The best evaluation accuracy for our experiment is obtained after 2 epochs of training.

Eval Acc: 89.55%



Leaderboard -



• Training and inference time

Training rate - 14.21batch/s

Evaluation rate - 45.25batch/s

Time taken for training 5 batches: 25.043 minutes

• Hyperparameters used in your experiment (e.g., number of epochs, learning parameter, dropout rate, hidden size of your model) and other details.

number of epochs - 5

learning parameter - gelu

dropout rate - 0.1

hidden size of your model - 3072

Batch size - 16

Learning rate - 0.0002

• Hypothesize what kinds of samples you might think your model would struggle with and report a minimum of ten incorrectly predicted test samples with their ground-truth labels. If you also report the confidence score of the predicted labels (the last Linear layer’s softmax score) on the samples, you will receive a bonus point.

Our model struggles to pick up the trends from sentences which have an implied negative meaning. For example, if steven soderbergh 's ` solaris ' is a failure it is a glorious failure . Also our model seems to have a high negative score on some words such as “eavesdropping”. Also the model struggles with some slangs. Also, our model fails to recognise third person references. eg, holden caulfield did it better.

Below are the sentences which were wrongly classified by our model.

| **idx** | **sentence** | **label** | **predicted label** | **score** |
| --- | --- | --- | --- | --- |
| 8 | you do n't have to know about music to appreciate the film 's easygoing blend of comedy and romance . | 1 | 0 | 0.6484504342 |
| 13 | we root for ( clara and paul ) , even like them , though perhaps it 's an emotion closer to pity . | 1 | 0 | 0.5315335393 |
| 22 | holden caulfield did it better . | 0 | 1 | 0.9743500352 |
| 62 | the primitive force of this film seems to bubble up from the vast collective memory of the combatants . | 1 | 0 | 0.8677293062 |
| 64 | the script kicks in , and mr. hartley 's distended pace and foot-dragging rhythms follow . | 0 | 1 | 0.6242036819 |
| 66 | if you 're hard up for raunchy college humor , this is your ticket right here . | 1 | 0 | 0.875779748 |
| 83 | though it 's become almost redundant to say so , major kudos go to leigh for actually casting people who look working-class . | 1 | 0 | 0.8440745473 |
| 87 | jaglom ... put ( s ) the audience in the privileged position of eavesdropping on his characters | 1 | 0 | 0.8301007748 |
| 92 | you wo n't like roger , but you will quickly recognize him . | 0 | 1 | 0.9699559808 |
| 93 | if steven soderbergh 's ` solaris ' is a failure it is a glorious failure . | 1 | 0 | 0.9998428822 |

• Potential modeling or representation ideas to improve the errors

Distilbert is a unigram model so using a ngram model would help capture better contextual meanings. Even though Distilbert uses pretraining to capture some relationships among the words but using a ngram model will surely enhance the capability to capture more context.

• Contribution section - please describe who did what

Programming Part - Peter Ortiz, [Isaac Blaine-Sauer](mailto:blain075@umn.edu)

Report and Analysis Part - Srijan Pal, Amitabha Deb

• (optional) What was the most challenging part of this homework?

Fine tuning the hyperparameters was a challenge as each run takes about 30 mins. Also understanding the datatype of the data used by our model was difficult.